Federal Communications Commission
Washington, D.C. 20554

EX PARTE OR LATE FILED

DOCKET FILE COPY ORIGINAL

May 13, 1998

RECEIVED

MAY 14 1998

FEDERAL COMMUNICATIONS COMMISSION
OFFICE OF THE SECRETARY

Richard N. Clarke
AT&T
Room 5462C2
295 North Maple Avenue
Basking Ridge, New Jersey 07920

Dear Mr. Clarke:

Attached please find an analysis prepared by staff members of the Common Carrier Bureau that deals with the HAI model's customer location algorithm. The analysis consists of a memorandum describing the analysis and a computer disk containing the results of the individual trials of the analysis. We would appreciate your comment on this analysis at your earliest convenience.

Sincerely,

Lisa S. Gelb

Lisa S. Gelb
Chief, Accounting Policy Division
Common Carrier Bureau

Enclosures

cc: BCPM sponsors
Office of the Secretary, CC Docket Nos. 96-45, 97-160

MEMORANDUM
UNITED STATES GOVERNMENT

Date: May 13, 1998

To: Magalie Roman Salas, Secretary

From: Jeffrey Prisbrey

Subject: A Test of Customer Dispersion in the HAI Customer Location Algorithm

On April 17, 1998 Sprint filed an ex parte containing an analysis of some customer location data inputs used in the HAI model for the state of Nevada. Sprint claimed that, for certain configurations of customer locations, the HAI model might understate the true cost of serving those customers because of the way in which the model converts the geocoded customer locations into a rectangular serving area. AT&T and MCI responded to the Sprint claims in an ex parte dated April 23, 1998, by claiming that "the situation identified by Sprint is rare, the total cost effect of correcting it is minor -- and likely to be at least partially offset by the effects of other HAI Model assumptions that have tended to overestimate costs."[1]

This memorandum documents the results of a test of the customer location algorithm used by the HAI model v5.0a that I have conducted as part of the ongoing staff evaluation of cost proxy models.[2] The test is designed to provide a measure of dispersion of customer locations both before and after an algorithm similar to the HAI algorithm is applied to create rectangular serving areas. Unlike the Sprint analysis, which relied on specific analysis of certain actual clusters in Nevada, the present analysis is based on a Monte Carlo simulation of a large number of randomly generated customer locations. This analysis does not attempt to evaluate the actual distribution or feeder algorithms used in the HAI model. Instead, it attempts to test the accuracy of the preprocessing algorithms used in converting geocoded and surrogate geocoded customer locations into rectangular serving areas that are used by the model to construct distribution and feeder plant.

I implemented the test using the MapBasic programming environment (a copy of the test's source code is included as Appendix 1 to this memorandum). Each trial of the test is made up of the following steps:

[1] MCI ex parte of April 23, 1998, page 1.

[2] Letter from Richard N. Clarke, AT&T, on behalf of HAI proponents AT&T and MCI, to Magalie Roman Salas, FCC, dated February 3, 1998 (CC Docket No. 96-45).

(1) I randomly generate a series of N points. The points all lie within a 18 kft by 18 kft box. The set of points represents a cluster of customer locations that could be served by a single SAI.

(2) I calculate two different measures of dispersion for the randomly generated points. The first measure is the total length of a Star Network. A Star Network is constructed by connecting each point in the cluster directly to the centroid of the cluster. The centroid of the cluster is represented by the average X and Y coordinates of the points. The second measure is the length of the Minimum Spanning Tree (MST).[3] The MST is the shortest possible connect-the-dots type graph that includes every point in the cluster. I do not want to imply that either measure is a good measure of the length of a cluster's distribution plant. Both measures, however, are measures of the dispersion of points.

(3) I apply the HAI algorithm's implications, as I then understood them.[4] I first calculate the area and aspect ratio of the convex hull of the points. I then find the height and width of a rectangular distribution area that has the same area and aspect ratio as the convex hull. Next, I find the dimensions of the individual customer's lots. Like HAI's algorithm, I constrain the dimensions of the customer's lots so that: (1) the area of N lots is equal to the area of the rectangular distribution area, and (2) the height of a lot is twice its width.

Once I have calculated these various dimensions, I "lay out" the lots in rows and columns. I make the first column by stacking lots end-to-end and North-to-South in the rectangle (assuming North-to-South is the long side). I stack lots on the first column until there is no more room in the rectangle; I allow the last lot to overflow the rectangle. I then add as few additional columns as possible. I only add columns until the total number of lots is just greater than or equal to the number of customers, N.

Once I have laid out the lots, I allocate customers as if they were in density zone 1 or

[3] For information about Minimum Spanning Trees and an algorithm used to find them, see Prim, R.C., November 1957, "Shortest Connection Matrix Network and Some Generalizations," *Bell System Technical Journal*: 36, 1389-1401.

[4] After further conversations with various HAI proponents, I am convinced that my test actually overstates the amount of dispersion implied by the HAI algorithm. The HAI algorithm does not actually determine the location of all individual lots, like I have done. The algorithm only determines the location of four lots, one in each corner of the rectangular distribution area. The HAI model then builds plant as if the customers were all located within the boundaries implied by these four lots. There is an internal difficulty here, because N lots of the prescribed dimensions often don't fit within those boundaries. I've ignored this problem and allowed the lots to overflow the boundaries. Because of this, my estimates of the dispersion implied by the HAI algorithm may be too high.

2.[5] I locate the first customer in the most South-Western lot, half-way along the lot's frontage (or width) and 150 feet South of it's Northern border. I add additional customers, each one lot width to the East, until there are no more lots in the row. I start the second row 300 feet North of the first row. I start the third row two lot heights North of the first row. I continue this pattern until I have located N customers, then I stop. If there are any unpopulated lots, and there typically are, they are in the Eastern part of the most Northern row.

If there are an odd number of North/South lots, I move the customers in the most Northern row from the top to the bottom of their lots.

(4) I calculate the length of the Star Network and of the MST for these new HAI points.

Please note that what I have described above is only a test of the accuracy of the customer locations used in the preprocessing stages of the HAI algorithm. It is not a test of the adequacy of the distribution plant subsequently built by the HAI model given those locations.

At this time, I have performed 2440 trials of this experiment. I have 15 trials for each N in [5,100] and 10 trials for each N in [101,200]. I have computed the error rate for each trial, a negative error meaning that HAI underestimated the dispersion. For example, in trials with N equal to 25, the HAI algorithm underestimates the length of the Star Network by an average of 15.4 percent. It underestimates the length of the MST by an average of 41.5 percent. These underestimates correspond to error rates of -0.154 and -0.415, respectively.

I have also made kernel smooths of the data. Kernel smoothing is a non-parametric regression technique that allows you to plot an underlying trend line given "noisy" data.[6] For simplicity, I used a uniform kernel to estimate the HAI algorithm's error rate as a function of N. The estimation process is equivalent to using an unweighted five period moving average. Figure 1 is a plot of the kernel smooths. The solid line represents the expected error rate, or bias, in the length of the Star Network, the broken line represents the expected bias in the length of the MST.

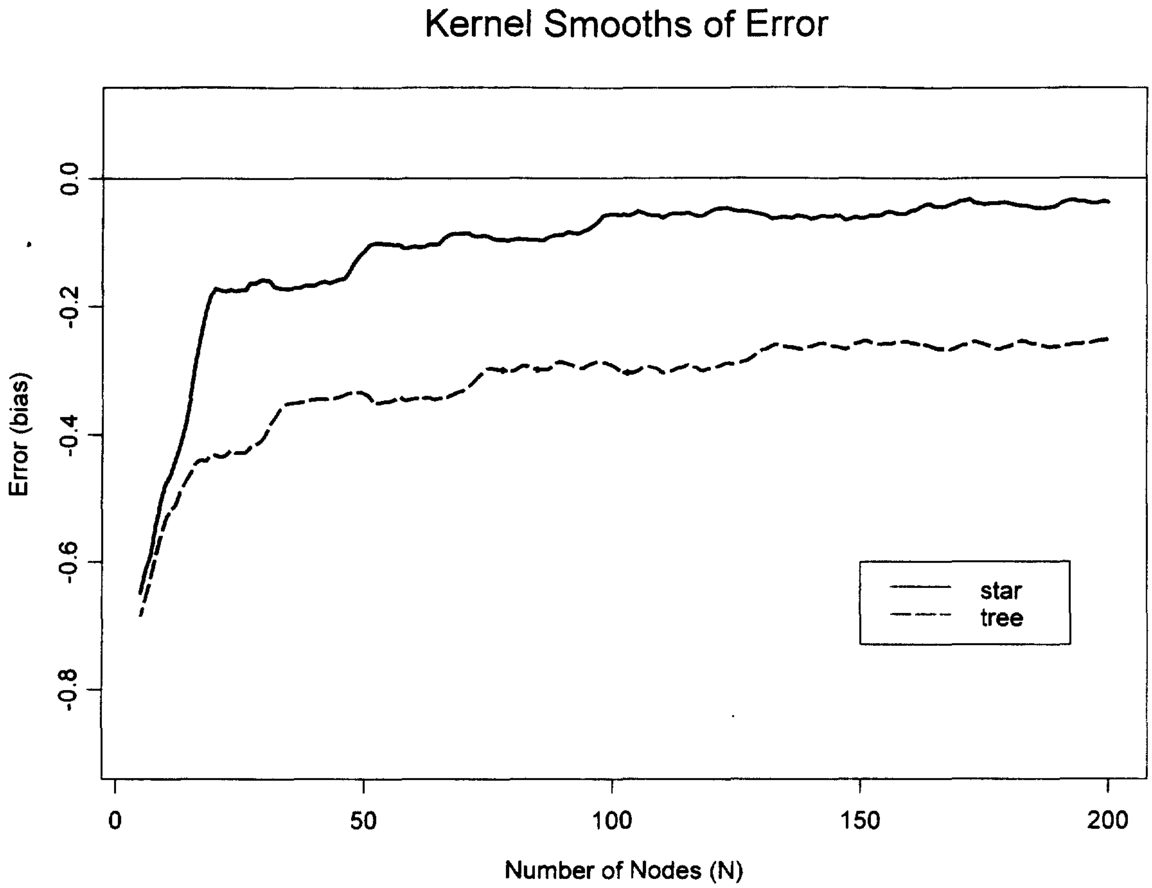
As the graph illustrates, the expected bias is negative over the whole range of N, for both measures. The bias seems to be a function of N, with the size of the bias increasing as N gets smaller. The shape of the curves suggests the following simple parameterization of the relationship between N and the error rate, E:

---

[5] The HAI model defines density zone 1 as an area with 0-5 lines per square mile, and density zone 2 as an area with 6-100 lines per square mile.

[6] For more information about non-parametric regression and smoothing see: Manski, C.F., March 1991, "Regression," *Journal of Economic Literature* XXIX: 34-50, and Härdle, W., 1989, Applied Nonparametric Regression, Cambridge: Cambridge University Press.

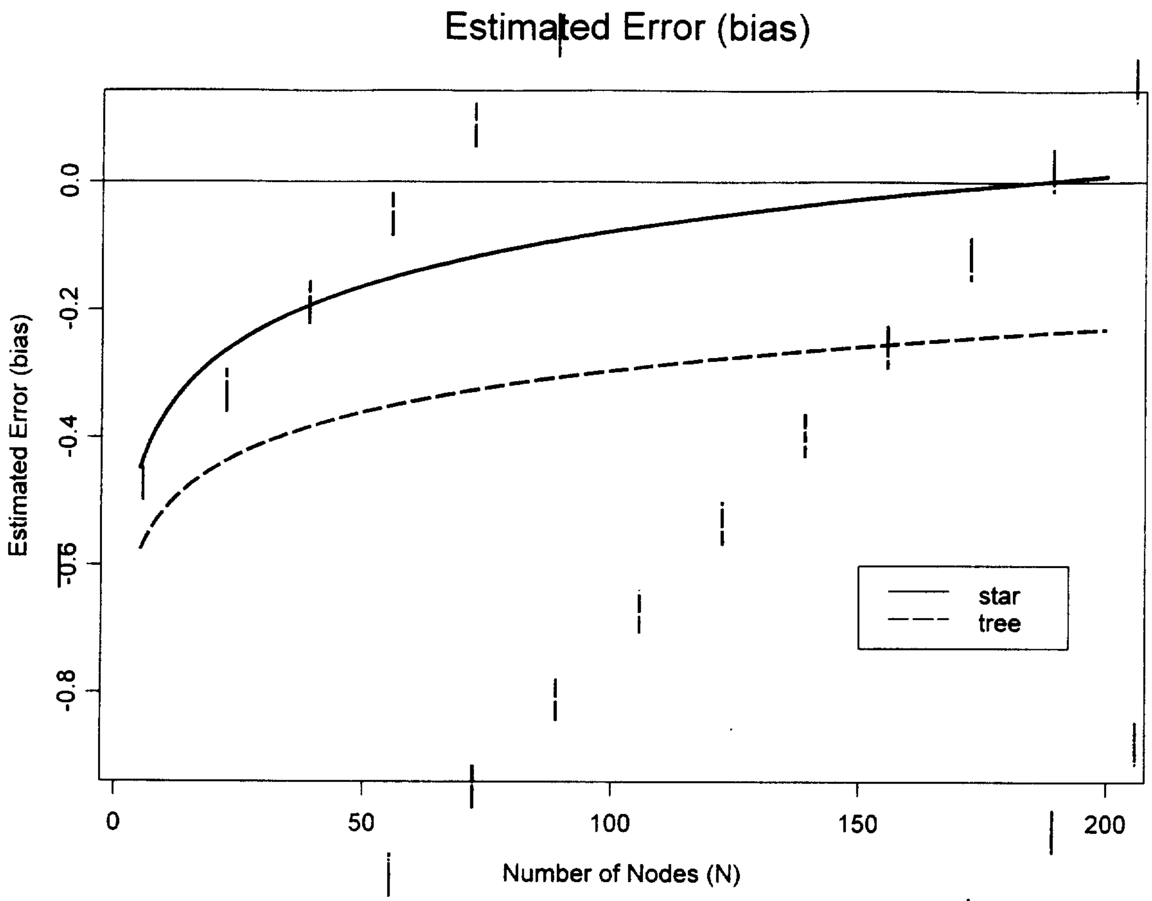
Figure 1:



$$E = b_1 + b_2 \ln(N) + \varepsilon,$$

where b_1 is an intercept parameter, b_2 is a slope parameter, and ε is a random disturbance. Using this parameterization, I've run regressions of the error rates under both measures. In both cases, the relationship between E and N was statistically significant, with p-values essentially equal to zero. The results of the regressions are in the Table 1. Figure 2 shows the curves implied by the estimation.

Figure 2:



Estimated Error (bias)
0.0
-0.2
-0.4
-0.6
-0.8
Estimated Error (bias)
0
50
100
150
200
Number of Nodes (N)
star
tree

Table 1:

Star	Value	Standard Error	t-stat	Prob (>|t|)
b1	-0.6471	0.0069	-94.0937	0.00
b2	0.1238	0.0016	78.2301	0.00
R-Squared	0.7151			
Residual Standard Error	0.06531			
df	2438			

MST	Value	Standard Error	t-stat	Prob (>|t|)
b1	-0.7268	0.0038	-193.5693	0.00
b2	0.0937	0.0009	108.4149	0.00
R-Squared	0.8282			
Residual Standard Error	0.03566			
df	2438			

Appendix 1: The MapBasic source code for running the test.

I performed my test in two series. The first series consisted of 5 trials for each N in [5,100]. You should be able to replicate this series by seeding the random number generator with -1. The second series consisted of 10 trials for each N in [5,200]. You should be able to replicate this series by seeding the random number generator with -2. Note: you can toggle interactive mode on and off before you compile.

```
'-- May 1, 1998.
'-- This program conducts a Monte-Carlo type test of HAI V5.0's Customer
'-- Location Algorithm, as I understand it.
'--
'-- Jeff Prisbrey
'-- Federal Communications Commission

Include "Mapbasic.Def"

Declare Sub Main
Declare Function Hull(X(),Y() as Float, HullInd(), NoPts, HullCnt as Integer) as Logical
Declare Function Tree(X(),Y() as float, ToNode(), N as integer) as Logical
Declare Function Rnd3(ByVal idum As Integer) As Float

'-- I need these global variables for the random number generator.
Global iff, inext, inextp, ma(55) As Integer


Sub Main
Close All
Dim X(300), Y(300), nX(300), nY(300), rX, rY as Float
Dim HullInd(300), HullCnt, ToNode(300), N as Integer
Dim NSlots, EWlots, Tlots as Integer
Dim i, j, k, t1, t2 as Integer
Dim starD1, starD2, treeD1, treeD2 as float
Dim cenX, cenY, Area1, AspectR, H, W, a, b as Float
Dim Result as Logical
Dim RegionObj as Object
Dim RectObj as Object

'-- Initialize Knuth's random number generator.
H = Rnd3(-2)
H = 0.0

'-- Create a place to store the results.
```

```
Create Table Results
  (Trial integer,
    NumNodes integer,
    ETime integer,
    StarDist1 float,
    StarDist2 float,
    TreeDist1 float,
    TreeDist2 float
    )
  File ApplicationDirectory$() + "ResultsB"
  Type DBF
Set Table Results
  FastEdit On
  Undo Off

Set Window Message Position (7,.75) Height 3.5
Set CoordSys Nonearth Units "ft" Bounds (-180000,-180000) (180000,180000)

N = 4
For j = 1 to 1960
  t1 = timer()
  starD1 = 0.0
  starD2 = 0.0
  treeD1 = 0.0
  treeD2 = 0.0

  '-- Create a place for the objects, and a coordinate system for plotting.
  If NumTables() > 1 Then
    If TableInfo(Cluster, Tab_Info_NRows) > 0 Then
      Drop Table Cluster
    End If
  End if
  Create Table Cluster
    (Id Char(10))
    File ApplicationDirectory$() + "Cluster"
  Set Table Cluster
    FastEdit On
    Undo Off
  Create Map For Cluster
    CoordSys Nonearth Units "ft" Bounds (-180000,-180000) (180000,180000)
  Map from Cluster
  Set Map Center (9000,9000) Layer 1 Editable On Selectable On

  '-- Increment N if needed, pick random points.
```

```
'-- Note that the points are within an 18 kft box and could therefore
'-- represent customer locations in a single cluster.
If ((j - 1) MOD 10) = 0 Then
  N = N + 1
End If
For i = 1 to N
  X(i) = Rnd3(1) * 18001
  Y(i) = Rnd3(1) * 18001
  Create Point(X(i), Y(i))
Next
Set Map Zoom Entire Layer 1

'-- Find the points which define the convex hull of the cluster.
'-- Create the convex hull and insert it into the table.
Result = Hull(X, Y, HullInd, N, HullCnt)
Create Region Into Variable RegionObj 0 Brush (1,0,0)
For i = 1 to HullCnt
 Alter Object RegionObj Node Add (X(HullInd(i)),Y(HullInd(i)))
Next
Insert Into Cluster (obj) Values (RegionObj)

'-- Find the centroid of the cluster.
cenX = 0.0
cenY = 0.0
For i = 1 to N
  cenX = cenX + X(i)
  cenY = cenY + Y(i)
Next
cenX = cenX / N
cenY = cenY / N

'-- Find the distance in the initial "Star" network.
'-- Find the distance in the inital minimum spanning tree.
Result = Tree(X, Y, ToNode, N)
starD1 = Distance(cenX, cenY, X(1), Y(1), "ft")
treeD1 = 0.0
For i = 2 to N
  treeD1 = treeD1 + Distance(X(i),Y(i),X(ToNode(i)),Y(ToNode(i)),"ft")
  Create Line (X(i) ,Y(i)) (X(ToNode(i)) , Y(ToNode(i)))
    Pen Makepen(1,2,RGB(255,0,0))
  starD1 = starD1 + Distance(cenX, cenY, X(i), Y(i), "ft")
Next

'-- HAI treats areas less than 0.03 square miles as high-rise buildings.
```

```
'-- I'll exclude consideration of these for now.
If Area(RegionObj, "sq mi") > 0.03 then

 '-- Compute the area and aspect ratio for the representative rectangle.
 Area1 = Area(RegionObj, "sq ft")
 AspectR = Distance(ObjectGeography(RegionObj,OBJ_GEO_MINX),
                ObjectGeography(RegionObj,OBJ_GEO_MINY),
                ObjectGeography(RegionObj,OBJ_GEO_MINX),
                ObjectGeography(RegionObj,OBJ_GEO_MAXY),
                "ft") /
         Distance(ObjectGeography(RegionObj,OBJ_GEO_MINX),
                ObjectGeography(RegionObj,OBJ_GEO_MINY),
                ObjectGeography(RegionObj,OBJ_GEO_MAXX),
                ObjectGeography(RegionObj,OBJ_GEO_MINY),
                "ft")

 '-- For simplicity, always make the rectangle taller than it is wide.
 If AspectR < 1 then
  AspectR = 1 / AspectR
 End If

 '-- The dimensions of the rectangular region.
 W = Sqr(Area1 / AspectR)
 H = W * AspectR

 '-- For plotting purposes, this is the lower corner of the rectangular region.
 rX = ObjectGeography(RegionObj,OBJ_GEO_MINX) - W * 3 / 2
 rY = ObjectGeography(RegionObj,OBJ_GEO_MINY)

 '-- Create the rectangular region and insert it into the table.
 Create Region Into Variable RectObj 0 Brush (1,0,0)
  Pen Makepen(1,2,RGB(0,0,255))
 Alter Object RectObj Node Add (rX,rY)
 Alter Object RectObj Node Add (rX,rY+H)
 Alter Object RectObj Node Add (rX+W,rY+H)
 Alter Object RectObj Node Add (rX+W,rY)
 Insert Into Cluster (obj) Values (RectObj)

'-- a is the width of a lot, b is the depth.
 a = Sqr(Area1/(2*N))
 b = 2*a

 '-- Determine the number of NS and EW lots
 NSlots = (H \ b) + 1
```

```
EWlots = ((N - 1) \ NSlots) + 1
Tlots = NSlots * EWlots

'-- Plot the lot regions.
For i = 0 to EWlots
  Create Line (rX + i*a, rY) (rX + i*a, rY + NSlots*b)
    Pen Makepen(1,2,RGB(0,255,0))
Next
For i = 0 to NSlots
  Create Line (rX , rY+ i*b) (rX + EWlots*a, rY + i*b)
    Pen Makepen(1,2,RGB(0,255,0))
Next


'-- Populate the lots, calculate customer locations.
'-- Start at location 1 and work your way east.  When you
'-- get to the end of the lots, move up a row.
'-- The variable k follows the rows.
nX(1) = rX + a / 2
nY(1) = rY + (b - 150)
For i = 2 to N
  nX(i) = nX(1) + a * ((i-1) MOD EWlots)
  k = (i-1) \ EWlots
  nY(i) = nY(1) + (k MOD 2) * 300 + ((k \ 2) * 2) * b
Next '-- i

'-- If the number of rows is odd, bring the locations in the top row
'-- down to the bottom part of their lots.
If (NSlots MOD 2) = 1 then
  For i = Tlots to (Tlots - EWlots + 1) Step -1
    nY(i) = nY(i) - b + 300
  Next
End If

'-- Find the centroid of the new points.
cenX = 0.0
cenY = 0.0
For i = 1 to N
  Create Point (nX(i), nY(i))
  cenX = cenX + nX(i)
  cenY = cenY + nY(i)
Next
cenX = cenX / N
cenY = cenY / N
```

```
  Set Map Zoom Entire Layer 1

  '-- Find the distance in the new "Star" network.
  '-- Find the distance in the new minimum spanning tree.
  Result = Tree(nX, nY, ToNode, N)
  starD2 = Distance(cenX, cenY, nX(1), nY(1), "ft")
  treeD2 = 0.0
  For i = 2 to N
    treeD2 = treeD2 + Distance(nX(i),nY(i),nX(ToNode(i)),nY(ToNode(i)),"ft")
    Create Line (nX(i) , nY(i)) (nX(ToNode(i)) , nY(ToNode(i)))
      Pen Makepen(1,2,RGB(255,0,0))
   starD2 = starD2 + Distance(cenX, cenY, nX(i), nY(i), "ft")
  Next

  '-- Write the results to the screen.
  Print Chr$(12)
  Print "Trial No. " + Str$(j)
  Print "Time for Trial: " + Str$(Timer() - t1)
  print "N: " + Str$(N)
  print " "
  print "starD1: " + Str$(starD1)
  print "starD2: " + Str$(starD2)
  print "%error: " + Str$(100*(starD2 - starD1)/starD1)
  print " "
  print "treeD1: " + Str$(treeD1)
  print "treeD2: " + Str$(treeD2)
  print "%error: " + Str$(100*(treeD2 - treeD1)/treeD1)

End If '-- Not a high-rise.

'-- Write the results to the Results table.
t2 = Timer() - t1
Insert Into Results (Trial,NumNodes,ETime,StarDist1,StarDist2,TreeDist1,TreeDist2)
  Values (j,N,t2,starD1,starD2,treeD1,treeD2)

'-- Switch the > or < sign to toggle interactive mode.
If N > 0 then
  Dialog Title " Do It Again? " Width 100 Height 50 Position 50,540
    Control OKButton Title "Again"
    Control CancelButton Title "Exit"
  If not CommandInfo(CMD_INFO_DLG_OK) Then
    Exit For
  End If
End If
```

```
Next '-- j
Close All
End Sub '-- Main


Function Hull(X(),Y() as Float, HullInd(), NoPts, HullCnt as Integer) as Logical
  '-- This function is taken from code published on the internet by The MapTools Company.
  '-- It's an implementation of Graham's Algorithm for finding the convex hull of a
  '-- set of points.

  ' The Hull function returns "False" if NoPts < 3 or for duplicate points
  '   or all the points on a line

  Dim Dist, T, Angle, D2MN, D2IM as Float
  Dim I, J, M, N, AngleInd as Integer
  Hull = False
  If NoPts < 3 Then Exit Function End If

   ' Find the pair M,N with the greatest separation
  Dist = 0
  For I = 1 to NoPts
    For J = I+1 to NoPts
     T = (X(J)-X(I))^2 + (Y(J)-Y(I))^2
     If T > Dist Then
       M = I   N = J   Dist = T
     End If
    Next
  Next
  If Dist = 0 Then Exit Function End If

   ' Find the rightmost point from the line M to N
  Angle = -1
  AngleInd = M
  For J = 1 to 2
    D2MN = ( (X(N)-X(M))^2+(Y(N)-Y(M))^2 )
    If D2MN = 0 Then Exit Function End If
    For I = 1 to NoPts
      If I <> M and I <> N Then
        D2IM = (X(I)-X(M))^2+(Y(I)-Y(M))^2
        If D2IM = 0 Then  Exit Function  End If
        T = ((X(I)-X(M))*(Y(N)-Y(M)) - (X(N)-X(M))*(Y(I)-Y(M)))/
          Sqr(D2IM * D2MN)
        If (T > Angle) Then Angle = T AngleInd = I End If
        If (T = Angle and (X(I)-X(M))^2 + (Y(I)-Y(M))^2 <
```

```
        (X(AngleInd)-X(M))^2 + (Y(AngleInd)-Y(M))^2) Then
        AngleInd = I End If
    End If
  Next
  ' Test to make sure some point is to the right,
  '    otherwise exchange M and N and do again
  If Angle > 0 Then Exit For End If
  If J = 2 then Exit Function End If
  AngleInd = N    N = M    M = AngleInd
 Next
 HullInd(1) = M    HullInd(2) = AngleInd   HullCnt = 2
 N = M   M = AngleInd

  ' Find the point at the greatest clockwise angle from current Hull edge
 Do
  Angle = 1   AngleInd = M
  D2MN = ( (X(N)-X(M))^2+(Y(N)-Y(M))^2 )
  For I = 1 to NoPts
   If I <> M and I <> N Then
    D2IM = (X(I)-X(M))^2+(Y(I)-Y(M))^2
    If D2IM = 0 Then Exit Function End If
    T = ((X(I)-X(M))*(X(N)-X(M)) + (Y(I)-Y(M))*(Y(N)-Y(M)))/
      Sqr(D2IM * D2MN)
    If (T < Angle) Then
      Angle = T  AngleInd = I  End If
    If (T = Angle and (X(I)-X(M))^2 + (Y(I)-Y(M))^2 <
      (X(AngleInd)-X(M))^2 + (Y(AngleInd)-Y(M))^2) Then
        AngleInd = I End If
   End If
  Next
  HullCnt = HullCnt + 1   HullInd(HullCnt) = AngleInd
  N = M    M = AngleInd
 Loop Until HullInd(HullCnt) = HullInd(1)
 Hull = True
End Function


Function Tree(X(), Y() as float, ToNode(), N as integer) as Logical
 '-- This is an implementation of Prim's minimum spanning tree algorithm.

 Dim inTree(300) as logical
 Dim minDist(300) as float
 Dim d, minD as float
 Dim i, j, k, m as integer
```

```
For i = 1 to N
  inTree(i) = False
  ToNode(i) = -1
  minDist(i) = 1.79769313486232E+307
Next
inTree(1) = True

j = 1
For i = 2 to N
  minD = 1.79769313486232E+307
  For k = 2 to N
    If (inTree(k) = False) then
      d = Distance(X(k),Y(k),X(j),Y(j),"ft")
      If d < minDist(k) then
        minDist(k) = d
        ToNode(k) = j
      End If
      If minDist(k) < minD then
        minD = minDist(k)
        m = k
      End If
    End If '-- inTree
  Next '-- k
  inTree(m) = True
  j = m
Next '-- i

 Tree = True
End Function


Function Rnd3(ByVal idum As Integer) As Float
'-- Subtractive Random Number Generator due to Knuth.  From Numerical Recipes.
'-- This is better than an LCR generator for doing Monte-Carlo type stuff.

Dim i, ii, k As Integer
Dim mj, mk As Integer

If (idum < 0) Or (iff = 0) Then
  iff = 1
  mj = 161803398 - Abs(idum)
  mj = mj Mod 1000000000
  ma(55) = mj
```

```
  mk = 1
  For i = 1 To 54
    ii = (21 * i) Mod 55
    ma(ii) = mk
    mk = mj - mk
    If (mk < 0) Then mk = mk + 1000000000 End If
    mj = ma(ii)
  Next
  For k = 1 To 4
    For i = 1 To 55
      ma(i) = ma(i) - ma(1 + (i + 30) Mod 55)
      If (ma(i) < 0) Then ma(i) = ma(i) + 1000000000 End If
    Next
  Next
  inext = 0
  inextp = 31
  idum = 1
End If

inext = inext + 1
If (inext = 56) Then inext = 1 End If
inextp = inextp + 1
If (inextp = 56) Then inextp = 1  End If
mj = ma(inext) - ma(inextp)
If (mj < 0) Then mj = mj + 1000000000 End If
ma(inext) = mj
Rnd3 = mj * 1.0 / 1000000000.0

End Function
```

Attachment A

DOCUMENT OFF-LINE

This page has been substituted for one of the following:

o An oversize page or document (such as a map) which was too large to be scanned into the RIPS system.

o Microfilm, microform, certain photographs or videotape.

o Other materials which, for one reason or another, could not be scanned into the RIPS system.

The actual document, page(s) or materials may be reviewed by contacting an Information Technician. Please note the applicable docket or rulemaking number, document type and any other relevant information about the document in order to ensure speedy retrieval by the Information Technician.

1 Diskette